

Tópico 05

Hugo Silva

Memória  
semicondutora

Circuito lógico

Correção de  
Erros

## Tópico 05 - Memória principal

Hugo Vinícius Leão e Silva

`hugovlsilva@gmail.com, hugo.vinicius.16@gmail.com, hugovinicius@ifg.edu.br`

Instituto Federal de Educação, Ciência e Tecnologia de Goiás  
Campus Anápolis  
Curso de Bacharelado em Ciência da Computação

2 de junho de 2023

Tópico 05

Hugo Silva

Memória  
semicondutora

Circuito lógico

Correção de  
Erros

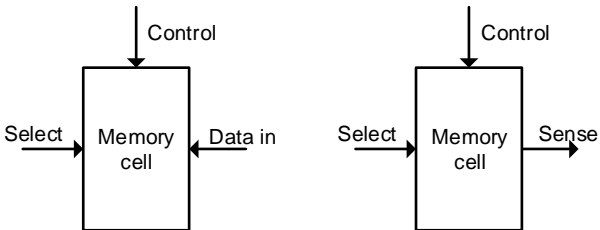
**1** Memória semicondutora

**2** Circuito lógico

**3** Correção de Erros

- O elemento básico de uma memória semicondutora é a **célula de memória**, que possuem as seguintes características:
  - Possuem dois estados estáveis ou semiestáveis – 0 e 1;
  - São capazes de serem gravadas para *setar*/definir o estado pelo menos uma vez;
  - São capazes de ler o estado.

Figura: Operações na célula de memória

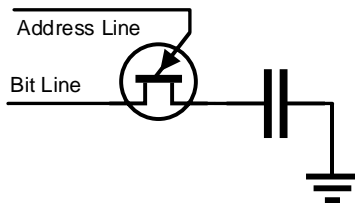


Os tipos de memória RAM possíveis são:

<b>Tipo</b>	<b>Categoria</b>	<b>Apagamento</b>	<b>Mecanismo de escrita</b>	<b>Volatilidade</b>
Random-access memory (RAM)	Leitura-escrita	Elétrico, nível de byte	Elétrico	Volátil
Read-only memory (ROM)	Leitura	Não aplicável	Máscara	Não volátil
Programmable ROM (PROM)	Leitura	Não aplicável	Elétrico	Não volátil
Eraseable PROM (EPROM)	Principalmente leitura	Luz UV, nível de chip	Elétrico	Não volátil
Electrically EPROM (EEPROM)	Principalmente leitura	Elétrico, nível de chip	Elétrico	Não volátil
Flash	Principalmente leitura	Elétrico, nível de bloco	Elétrico	Não volátil
3D Cross-Point (3D XPoint)	Leitura-escrita	Elétrico, nível de bit(?)	Elétrico	Não volátil

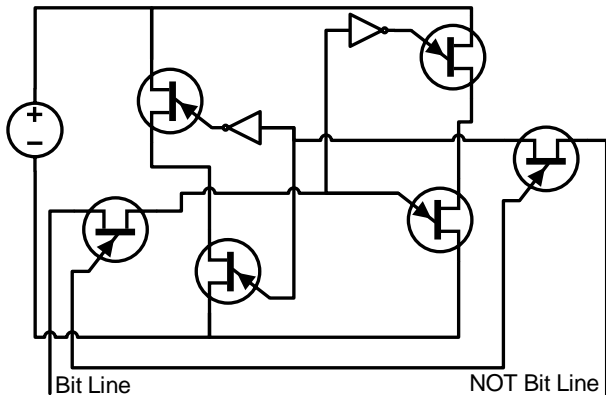
Memória RAM pode utilizar duas tecnologias, **DRAM** (RAM Dinâmica) e **SRAM** (RAM Estática). Na DRAM:

- DRAM é um dispositivo analógico;
- As células são capacitores. O valor de tensão é interpretado como 1 ou 0;
- Capacitores se descarregam mesmo quando energizados – DRAMs necessitam um *refresh* periódico para manter os dados.



## SRAM:

- SRAM é um dispositivo puramente digital – *flip-flops*;
- O bit é armazenado na célula enquanto ela for energizada – não é necessário *refresh periódico*



- DRAM é mais densa do que a SRAM;
- DRAM é mais barata do que a SRAM (principalmente para grandes quantidades de memória);
- DRAM exige um circuito de refresh; SRAM, não;
- DRAM é mais lenta do que a SRAM;
- DRAM é usada para memória principal; SRAM, para cache (mas há exceções).

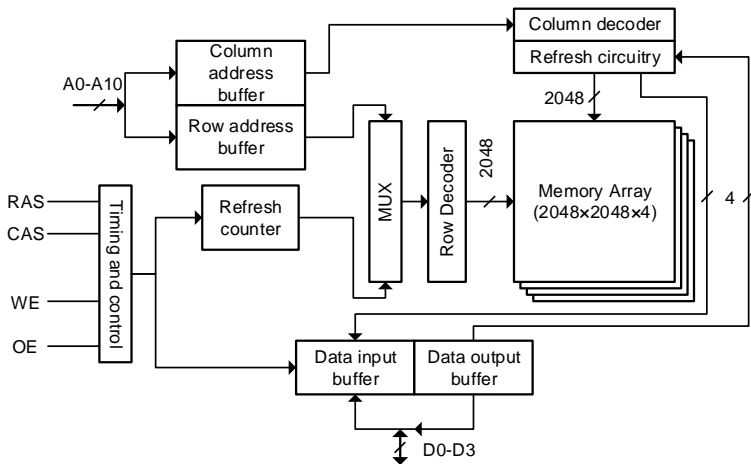
Memória ROM pode utilizar quatro tecnologias:

- ROM – Os dados são gravados **durante** o processo de fabricação do *chip* (sem chance para erros!);
- PROM – Os dados são gravados **uma única** vez (WORM – *Write Once, Read Many*) **depois** da fabricação do *chip*;
- EPROM – Os dados são gravados depois da fabricação do chip, que pode ser **completamente apagado** expondo o seu *die* à **luz UV**;
- EEPROM – Os dados são gravados depois da fabricação do chip, que pode ser **completamente apagado eletricamente**;
- Flash – Os dados são gravados depois da fabricação do chip, que pode ter **alguns blocos apagados eletricamente em alta velocidade**.



- Um *chip* contém um arranjo de células de memória visto como  $W$  palavras de  $B$  bits cada; ex.: um chip de 16 Mb pode ser uma memória de 1M-palavras de 16 bits ou 16M-palavras de um bit ou **4M-palavras de quatro bits**;
- Cada bit fica em uma matriz;
- $16 \text{ Mb} = 16.777.216 \text{ bits} \rightarrow \frac{16}{4} \text{ Mbits} = 4.194.304 \text{ bits} = 2048^2 \rightarrow \log_2 2048 = 11 \text{ bits}$  para endereçar linhas e colunas. As linhas de endereço são multiplexadas para economizar o n.º de pinos;
- Podem ser necessários vários chips de memória para casar com a largura do barramento de memória.

**Figura:** Diagrama de uma memória de 16 Mb ( $4M \times 4$ )



Uma operação de **leitura** em um chip genérico de RAM é realizado da seguinte forma:

- 1 Os sinais  $\overline{OE} = 0$  e  $\overline{WE} = 1$  são apresentados no *bus* **juntamente** com o endereço de linha em A0–A10 e com sinal  $\overline{RAS} = 0$  (*Row Address Select*);
- 2 Alguns CLKs depois,  $\overline{RAS} = 1$  e o sinal  $\overline{CAS} = 0$  (*Column Address Select*) é apresentado no *bus* **juntamente** com o endereço de coluna em A0–A10;
- 3 Alguns CLKs depois os dados lidos são apresentados no *bus* – esse atraso é conhecido como (*CAS Latency*).

Uma operação de **gravação** em um chip genérico de RAM é realizado da seguinte forma:

- 1  $\overline{OE} = 1$  e  $\overline{WE} = 0$ , o endereço de linha em A0–A10 e  $\overline{RAS} = 0$ ;
- 2 Alguns CLKs depois,  $\overline{RAS} = 1$ ,  $\overline{CAS} = 0$  e o endereço de coluna em A0-A10;
- 3 Alguns CLKs depois os dados a serem gravados são apresentados no *bus*.

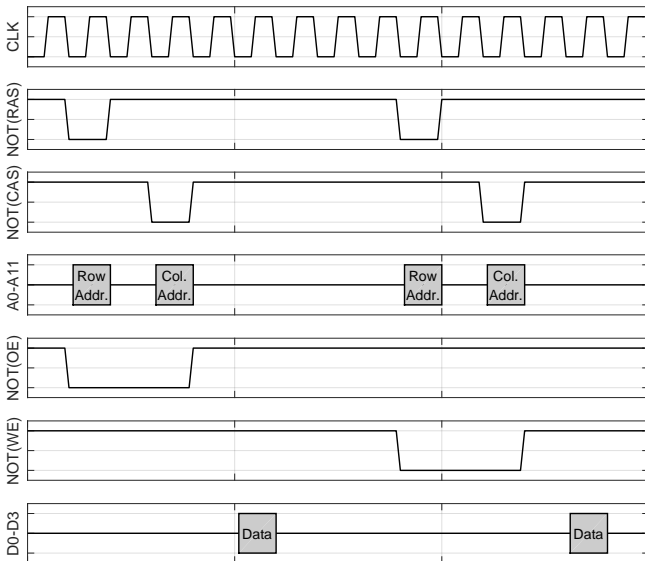
## Tópico 05

Hugo Silva

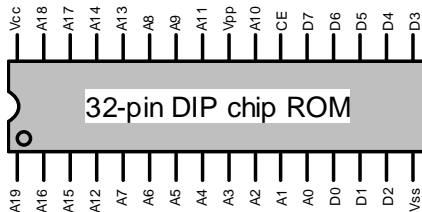
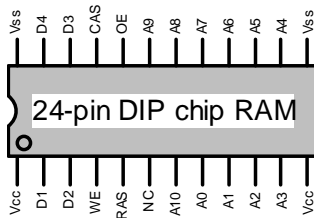
Memória  
semicondutora

Circuito lógico

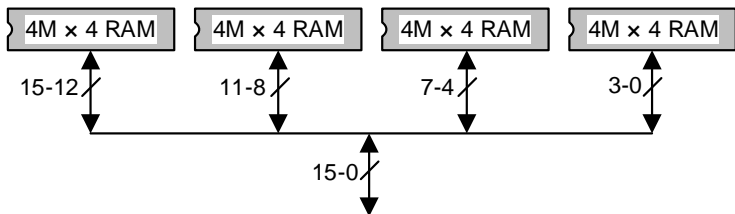
Correção de  
Erros



**Figura:** ROM e RAM com DIP *packaging*



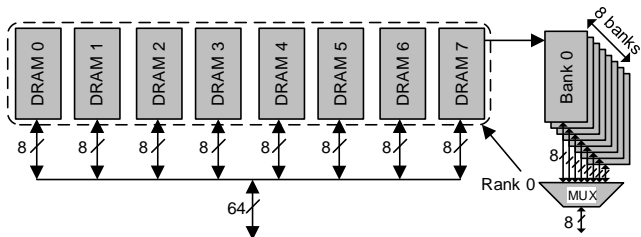
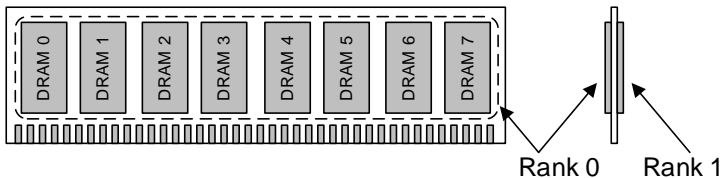
Vários *chips* DRAM podem/devem ser agrupados em paralelo para casar com a largura do barramento, formando um *memory rank*:



- Um *memory bank* é um grupo de *memory arrays* independentes;
- *Interleaving* é uma técnica que permite que os  $B$  bancos atendam a  $B$  operações independentemente → aumento das taxas de transferência/dados (*data rate*);
- A frequência de operação do *interleaving* é maior do que a frequência de cada banco individual;
- O circuito de controle utiliza o *round-robin* para alternar entre os bancos;
- Exemplo: cada banco de memória transfere dados a cada 10 ns; dois bancos: 5 ns (2x); quatro bancos: 2,5 ns (4x);
- *Memory banks*  $\neq$  *memory arrays*.



DIMM (*Dual-Inline Memory Module*) com dois *ranks* e oito *banks*:



Memory ranks são alternados utilizando o sinal  $\overline{CS}$  (*Chip Select*).

Um DIMM de 8 GB pode ter as seguintes configurações:

	<b>DIMM 1</b>	<b>DIMM 2</b>
N.º de <i>memory ranks</i>	1	2
N.º de DRAMs	8	16
N.º de <i>memory banks</i> em cada DRAM	8	8
Organização de cada DRAM	1G × 8 bits	512M × 8 bits
Densidade de cada DRAM	8 Gb	4 Gb
Organização da DIMM	1G × 64 bits	1G × 64 bits

- DRAM está sujeita a erros: **hard error** e **soft error**;
- *Hard error* é um erro permanente/físico e células de memória, que não são capazes de armazenar dados confiavelmente → uso intenso/severo; defeitos de fabricação; desgaste;
- *Soft error* é proveniente de erro aleatório e não danifica as células de memória → decaimento radioativo, radiação de fundo (raios cósmicos e nêutrons);
- São necessários processos de detecção e, se possível, correção de erros;
- Para isso, utilizam-se códigos corretores de erros (ECC - *Error-Correcting Codes*);
- Armazenam-se a palavra de  $M$  bits de dados e o código de  $K$  bits (cada palavra na memória ocupa  $(M + K)$  bits no total).

## Como funcionam os ECCs na leitura?

- Leem-se os  $(M + K)$  bits. Os  $M$  bits são comparados com os  $K$  bits do código;
  - 1 Nenhum erro foi encontrado? Os  $M$  bits são transferidos;
  - 2 Foi encontrado um erro corrigível? Os  $(M + K)$  bits são postos em um corretor e os  $M$  bits corretos são enviados;
  - 3 Foi encontrado um erro incorrigível? O erro é reportado.

## Como funcionam os ECCs na gravação?

- Obtêm-se os  $M$  bits a serem gravados, que são passados em um gerador de código de erro com  $K$  bits;
- Armazenam-se os  $(M + K)$  bits.

Desejamos codificar a palavra 1100 ( $M = 4$ ). Como codificar essa palavra usando o Código de Hamming?

- 1  $K$  deve respeitar à inequação  $2^K - 1 \geq M + K$ . Para  $M = 4$ ,  $K = 3$  e, portanto, *overhead* de codificação de  $\frac{7}{4} = 75\%$ ;
- 2 Complete a tabela para construir a **palavra codificada**:

Posição do bit		1	2	3	4	5	6	7	
		001	010	011	100	101	110	111	
Paridade ou dado?		P1	P2	D1	P3	D2	D3	D4	
Valor do bit		?	?	1	?	1	0	0	
Bits avaliados	P1	X		X		X		X	$P1 = D1 \oplus D2 \oplus D4$
	P2		X	X			X	X	$P2 = D1 \oplus D3 \oplus D4$
	P3				X	X	X	X	$P3 = D2 \oplus D3 \oplus D4$

- 3 Bits de dados: 1100; bits do código: 011;
- 4 A palavra final será na ordem 0111100.

Agora desejamos decodificar a palavra 0111100:

- 1 Lembre-se que  $M = 4$  e  $K = 3$ ;
- 2 Utilize a tabela para construir a **palavra síndrome**:

Posição do bit		1	2	3	4	5	6	7	
		001	010	011	100	101	110	111	
Paridade ou dado?		P1	P2	D1	P3	D2	D3	D4	
Valor do bit		0	1	1	1	1	0	0	
Bits avaliados	S1	X		X		X		X	$S1 = P1 \oplus D1 \oplus D2 \oplus D4$
	S2		X	X			X	X	$S2 = P2 \oplus D1 \oplus D3 \oplus D4$
	S3				X	X	X	X	$S3 = P3 \oplus D2 \oplus D3 \oplus D4$

- 3  $S1 = 0$ ,  $S2 = 0$  e  $S3 = 0$ ;
- 4 A palavra síndrome é codificada na ordem  $\{S3, S2, S1\}$ ;
- 5 Neste caso, a palavra síndrome é: 000.
  - Se a palavra síndrome só tem 0's, não tem erro;
  - Se houver apenas um bit 1, apenas os bits de paridade estão errados, nada precisa ser feito;
  - Se mais de um bit for 1, então o valor numérico da palavra síndrome indica o bit errado.

(cont'd)

- Outro exemplo: uma palavra de  $M = 8$  bits requer uma palavra síndrome de  $K = 4$  bits para corrigir erros **simples**, com *overhead* de codificação de  $\frac{12}{8} = 50\%$ :

$$(2^4 - 1) \geq (8 + 4) \rightarrow 15 \geq 12$$

- Observe que o *overhead* diminui à medida que  $M$  cresce:

	Single-Error Correction		Single-Error Correction/ Double-Error Detection	
Data Bits	Check Bits	% Increase	Check Bits	% Increase
8	4	50	5	62.5
16	5	31.25	6	37.5
32	6	18.75	7	21.875
64	7	10.94	8	12.5
128	8	6.25	9	7.03
256	9	3.52	10	3.91

- Nota: existem códigos SEC (*single-error-correcting*) e SEC-DED (*single-error-correcting, double-error-detecting*).



Tópico 05

Hugo Silva

Memória  
semicondutora

Circuito lógico

Correção de  
Erros

Capítulo abordado: 5